

[MDPI](#) [Download PDF \(/1424-8220/23/13/6117/pdf?version=1688376187\)](#) [Share](#) [Order Article Reprints \(/1424-8220/23/13/6117/reprints\)](#)

[Introduction](#) [Literature Survey \(/about/journals\)](#) [Topics \(/topics\)](#) [Information \(/authors\)](#) [Author Services](#) [Toggle desktop layout cookie](#) [Q](#)

[Motivation of the Research](#) [Help \(/authors/english\)](#) [Initiatives \(/about/initiatives\)](#) [About \(/about\)](#)

[Methodology](#) [Results and Discussion](#) [Conclusions](#) [Cite](#) [Sign In / Sign Up \(/user/login\)](#)

[Author Contributions](#) [Funding](#) [Submit \(https://susy.mdpi.com/user/manuscripts/upload?journal=sensors\)](#)

[Institutional Review Board](#) [SciProfiles](#) [Statement](#) [groups/public/10.3390/s23136117?utm_source=mdpi.com&utm_medium=publication&utm_campaign=discuss_in_sciprofiles\)](#)

[Data Availability Statement](#) [Keyword](#) [Acknowledgments](#) [Endorse](#) [Affiliation / Email](#)

[Conflicts of Interest](#) [References](#) [Sensors](#) [Comment](#)

All Article Types

Search

Advanced Search

[Journals \(/about/journals\)](#) / [Sensors \(/journal/sensors\)](#) / [Volume 23 \(/1424-8220/23\)](#) / [Issue 13 \(/1424-8220/23/13\)](#) / [10.3390/s23136117](#) /

**sensors**[\(/journal/sensors\)](#)

Submit to this Journal
https://susy.mdpi.com/user/manuscripts/upload?form%5Bjournal_id%5D%3D3

Review for this Journal
<https://susy.mdpi.com/volunteer/journals/review>

Propose a Special Issue
[\(/journalproposal/sendproposalspecialissue/sensors\)](https://susy.mdpi.com/journalproposal/sendproposalspecialissue/sensors)

Article Menu[Back to Top](#)

Article Menu

[Download PDF \(/1424-8220/23/13/6117/pdf?version=1688376187\)](#)
[Introduction](#)
[Academic Editors](#)
[Motivation of the Research](#)
[Methodology](#)
[Results and Discussion](#)
[Conclusions](#)
[Author Contributions](#)
[Funding](#)
[Institutional Review Board](#)
[Supplementary Material](#)
[Data Availability Statement](#)
[Recommended Articles](#)
[Acknowledgments](#)
[Conflicts of Interest](#)
[References](#)
[More by Authors Links](#)
Article Views 1979

Citations 11

Table of Contents
[Open Access](#) [Article](#)

Algorithmic Approach to Virtual Machine Migration in Cloud Computing with Updated SESA Algorithm

 by **Amandeep Kaur**¹, **Saurabh Kumar**¹,

Deepali Gupta¹  (<https://orcid.org/0000-0002-3207-5248>),

Yasir Hamid²  (<https://orcid.org/0000-0003-1334-2651>),

Monia Hamdi³  (<https://orcid.org/0000-0003-3690-9868>),

Amel Ksibi⁴  (<https://orcid.org/0000-0001-6019-5960>),

Hela Elmannai^{3,*}  (<mailto:hseilmannai@pnu.edu.sa>)  (<https://orcid.org/0000-0003-2571-1848>) and

Shilpa Saini^{5,*}  (<mailto:shilpa.saini211986@gmail.com>)

[Back to Top](#)

1. [Download PDF \(/1424-8220/23/13/6117/pdf?version=1688376187\)](#)
[140401, India](#) [Share](#) [Order Article Reprints \(/1424-8220/23/13/6117/reprints\)](#)
2. [Literature Survey](#) [Abu Dhabi Polytechnic, Abu Dhabi](#) [Help](#)
3. [Methodology](#) [Department of Information Technology, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia](#) [Cite](#)
4. [Author Contributions](#) [Department of Information Systems, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia](#) [SciProfiles](#)
5. [Institutional Review Board Statement](#) [\(https://sciprofiles.com/discussion-public/10.3390/s23136117?\)](#)
- * [Informed Consent Statement](#) [medium=publication&utm_campaign=discuss_in_sciprofiles\)](#)
- [Data Availability Statement](#)
- [Acknowledgments](#)
- [Conflicts of Interest](#)
- [References](#)

Submission received: 6 March 2023 / Revised: 25 June 2023 / Accepted: 25 June 2023 / Published: 3 July 2023

(This article belongs to the Section **Intelligent Sensors** ([/journal/sensors/sections/Intelligent_Sensors](#)))

Download ✓

Browse Figures

(https://pub.mdpi-res.com/sensors/sensors-23-06117/article_deploy/html/images/sensors-23-06117-g001.png?1688376415)

(https://pub.mdpi-res.com/sensors/sensors-23-06117/article_deploy/html/images/sensors-23-06117-g002.png?1688376413)

(https://pub.mdpi-res.com/sensors/sensors-23-06117/article_deploy/html/images/sensors-23-06117-g003.png?1688376414)

(https://pub.mdpi-res.com/sensors/sensors-23-06117/article_deploy/html/images/sensors-23-06117-g004.png?1688376411)

Review Reports ([/1424-8220/23/13/6117/review_report](/1424-8220/23/13/6117/review_report))

Versions Notes (</1424-8220/23/13/6117/notes>)

[Back to Top](#)


[Download PDF \(/1424-8220/23/13/6117/pdf?version=1688376187\)](#)

Abstract

[Order Article Reprints \(/1424-8220/23/13/6117/reprints\)](#)

Cloud computing plays an important role in every IT sector. Many tech giants such as Google, Microsoft, and Facebook, as well as small businesses, are deploying their data centres around the world to provide cloud-based computing and storage services. The customers either submit their job directly or they take the help of the service providers for the submission of the jobs to the cloud centres. The preliminary aim of this paper is to survey the overall power consumption which was ignored in the early days of cloud computing. This was due to the performance expectations from cloud servers as they were supposed to provide all the services through their services layers IaaS, PaaS, and SaaS. As time passed, various research groups came up with new terminologies and algorithmic architecture for the management of power consumption and sustainability, other algorithmic anarchies were also introduced. Current state-of-the-art oriented learning and optimization algorithms. In this paper, an in-depth analysis is done on multiple approaches for migration among virtual machines and how they address various issues among existing approaches. The proposed work utilizes elastic scheduling inspired by the smart elastic scheduling algorithm (SESA) to develop a more efficient VM allocation and migration algorithm. The proposed work uses cosine similarity and bandwidth utilization as additional utilities to improve the current performance in terms of QoS. The proposed work is evaluated for overall power consumption and service level agreement violation (SLA-V) and is compared with related state of art techniques. A proposed algorithm is also presented in order to solve problems found during the survey.

Keywords: cloud computing (/search?q=cloud+computing); virtual machine (/search?q=virtual+machine); migration (/search?q=migration); power consumption (/search?q=power+consumption); SESA (/search?q=SESA)

1. Introduction

Cloud computing is one of the most emerging fields in modern-day developments. A cloud network is comprised of three layers of services as follows:

- (a) Infrastructure as a service (IaaS);
- (b) Platform as a service (PaaS);
- (c) Software as a service (SaaS).

To speed up the computation efficiency, the physical machines (PMs) (machines with physical attributes) are supported with virtual machines (VMs). There are two issues in the association of VM to the PM.

- (a) Allocation of a new VM to the PM;
- (b) Management of existing allocated VMs.

[Back to Top](#)

- Identify the applicable funding agency here. If none, delete this text box.

Both processes consume power and, hence, if they are not managed well, high power consumption will be observed. The world is already suffering from global warming and, hence, in such a scenario, high power consumption is not affordable. From the records, it is analyzed that one data centre consumes as much energy as 25,000 households [1]. Energy consumption is integral power over a given unit of time and can be defined by Equation (1)

- Conclusions

- Author Contributions

- Funding

where P is consumed power and dt is the time interval when the power is consumed which satisfies the quality of service (QoS) requirements of users considering the service level agreement (SLA). An efficient cloud model is also termed a green cloud. Green cloud architecture has four elements, as shown in Figure 1.

- Acknowledgments

- Conflicts of Interest

- References

- Comment

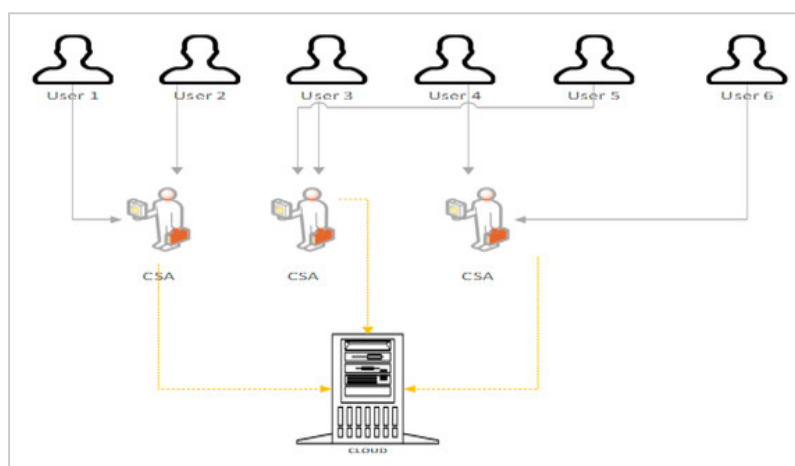


Figure 1. Green Cloud Components.

- Consumer or broker: A consumer or customer of the cloud submits its requirement directly to the cloud or gets it submitted by a broker;
- Cloud-service allocator (CSA): The cloud infra is not directly associated with the user and, hence, the CSA negotiates on SLA, prices, and other terms between the service provider and the customer. The service allocation has a service scheduler associated with a CSA, which deals with the completion time and scalability of customer demand;
- Physical machine (PM);
- Virtual machine (VM).

VM allocation has two scenarios, namely the admission of a new VM and migrations of one VM to another PM. The first scenario is observed at the preliminary stage of the cloud server initialization. The most basic architecture was proposed by Dr. Rajkumar Buyya in 2010, which is named modified best fit decreasing utilization (MBFD) which is an extension of the best fit decreasing algorithm (BFD). BFD was developed by Baker et al. in the late 1980s

Back to Top

and was used further in the early stages of cloud and grid computing [2]. When a VM is to be migrated from one PM and is to be allocated to another PM, two issues were addressed.

- Literature Survey
 - (a) Spot detection: Which PM is to be detected for the migration?
 - Motivation of the Research
 - (b) Destination PM detection: Where to migrate?
 - Methodology
 - The MBFD algorithm uses CPU utilization in order to allocate the VMs at the preliminary stages, i.e., when a new VM is to be allocated to the PM. The MBFD algorithm sorts the CPU utilization in descending order and looks for the resources with the highest CPU utilization containing VM. The algorithm also checks the most feasible host with the least power consumption. The pseudocode for MBFD is in Algorithm 1 as follows.
 - Results and Discussion
 - Conclusions
 - Author Contributions
 - Funding
 - Institutional Review Board Statement
 - Informed Consent Statement
 - Data Availability Statement
 - Acknowledgments
 - Conflicts of Interest
 - References
- (d) Calculate P_c
- (e) Check if P_c is least
- (f) If Yes, Allocate VM to Host
- (g) Reduce Host resources by the amount which is consumed by the VM
- (h) Pick Next VM

As time passed on, amendments have been monitored in MBFD. In 2015 Lu, X and Zhang [3] presented an enhanced MBFD algorithm for allocation, which considered load over the PMs. Lu and Zhang neutralized bandwidth, random access memory (RAM), and CPU utilization as follows.

$$\text{Normalized Demand (ND)} = w_1 \times \text{Ram} + w_2 \times \text{CPU} + w_3 \times \text{Bandwidth} \quad (2)$$

where w_1 , w_2 , and w_3 are weights for RAM, CPU, and bandwidth of the VM.

The modified MBFD algorithm follows MBFD by replacing the CPU utilization by ND. It also adds up a hot migration policy in which a threshold is set considering all the loads over the PMs and, if the PM exceeds the threshold, then VMs are selected for the migration. The load calculation is done by applying three conditions, as given in Equation (3). The equations considered host CPU utilization (HCPU), average CPU utilization (ACPU), host RAM utilization (HRAMU), average RAM utilization (ARAMU), and host bandwidth utilization (HBU), along with average bandwidth utilization (ABU) for overloaded migrations.

$$0 \quad \text{Otherwise} \\ 1 \quad \text{if } \text{HCPU} > \text{ACPU} \parallel \text{HRAMU} > \text{ARAMU} \parallel \text{HBU} > \text{ABU} \quad (3)$$

Back to Top

- **Abstracting load as an effective parameter.** Mann et al. presented a computation core-aware load-based MBFD algorithm, which also considered the CPU cores involved in the allocation process. In order to enhance power management, the distribution of load is minimized among the individual cores, and core-to-core migration is also incorporated [4].
- **Methodology of natural computing is also observed in MBFD advancements.** As per “de Casteleira and Discussion”, computing is a terminology which encompasses the algorithmic architecture which takes inspiration from nature or that employs natural phenomenon to complete a given task [5]. Cloud computing uses this phenomenon in order to identify the PM where VMs are to be migrated. Kansal et al. [6] used swarm intelligence (SI) for VM selection (swarm intelligence)-based behaviours observed in groups. For example, the behaviour of an ant colony, the behaviour of cuckoos and fireflies, etc.). The PM from where the VMs are to be migrated is chosen from the source nodes. The algorithmic architecture is further extended to use an Artificial Bee Colony (ABC) for the selection of the destination node where the VMs are to be migrated. The source node is selected by the computation of the “Attraction Index (AI)”, which is computed by the evaluation of “Consumed Energy (CE)” as shown in Equation (4).

$$CE = \frac{(\sum_{i=1}^n \sum_{j=1}^k CPU_{ij}) ((\sum_{i=1}^n \sum_{j=1}^k MU_{ij}))}{M} \quad (4)$$

where n is the total amount of VM executing over PM and k denotes the total amount of jobs allocated to i th VM. M corresponds to the total amount of memory units executing on the i th VM. In addition to CE , the computation time of the node or NCT corresponds to the alive node to compute the total invested time in order to complete supplied jobs is also calculated. NCT formulation is given by Equation (5).

$$NCT_{active} = \sum_{i=1}^n \sum_{j=1}^k NCT_{ijactive} \quad (5)$$

where NCT_{ij} is the execution time for k number of jobs supplied to n number of VMs. Using CE and NCT , AI is calculated by Equation (6).

$$AI_i = AI_{active}(CE_{active}, NCT_{active}) \quad (6)$$

By sorting the AI in decreasing order, the lowest AI index will be selected as the PM from which the VMs are to be migrated. In order to choose the VMs to be migrated, the load on each VM is calculated by using Equation (7).

$$Load_{ij} = \frac{\sum_{i=1}^n job_i}{\int_0^t Pc dt} \quad (7)$$

- VMs with a high load compared to the average load are considered to be migrated from the active PM. Durgadevi et al. [7] also used swarm-based hybridization of leapfrog search and cuckoo search for the VM and PM selection for migration. In addition to CPU utilization, load and computation time, the authors introduced entropy for the computation of VMs to be migrated.

Motivation of the Research

Methodology

- When it comes to selecting the VMs for the migration, several research articles have focused on VM selection that is based on threshold [1,3,6,7,8]. In order to detect the hotspot,

Belodlazov et al. proposed the minimization of migrations (MM). MM has two thresholds, namely upper and lower. If the usage of the host CPU is lower compared to the lower

threshold, all host VMs are migrated and if the utilization of the host CPU is higher than the upper threshold, some VMs are migrated through the PM. The working is illustrated in

Figure 2.

Figure 2. Working of the proposed algorithm.

Algorithm 2

Algorithm 2: VM Migration

1. Initialize VMList, HostList, and MigrationList.

2. For each host h in HostList do

3. If vm.utilization() > hUtil - upper threshold

4. Evaluate t as $t \leftarrow \text{vm.utilization()} - \text{hUtil} + \text{upper threshold}$

5. Keep vm as best fit vm until greater t is not attained or host's utilization does not go below upper threshold

6. Reduce hUtil by best fit vm.utilization()

7. Add vm to migration list

In this work, the main issue of power consumption, SLA-V, and optimal resource utilization has been addressed by allocating VM and migrating to the best available host in an energy-efficient manner. It has been done by developing an energy-efficient model using a modified SESA algorithm in order to improve the efficacy of the proposed model. Two additional parameters such as cosine similarity and bandwidth utilization have been employed.

2. Literature Survey

Masdari and Khezri (2020) introduced a detailed analysis of the process of VM migration [9]. Dubey and Sharma (2020) worked for the betterment of the utilization of the resources by introducing intelligent water drop (IWD) for the selection of the VMs to be migrated [10]. To check the effectiveness of the proposed algorithm, the authors increased the load through load files which are globally available on **cloudbus.org** (<http://cloudbus.org>) (accessed on 24 June 2023).

Joshi and Munisamy (2020) introduced the dynamic degree balance (DDB) in collaboration with MBFD. The authors also considered the load mechanism in contrast to the

VM allocation mechanism and, hence, the two policies run simultaneously to point the VM to an existing PM. The authors compared the algorithm with two static algorithms such as shortest job first (SJF) as well as first come first serve (FCFS). The rate of imbalance and waiting time to experiment have been considered as evaluation parameters and work simulation has been done by CloudSim [11].

Ruan et al. (2019) described a model for allocating VMs including migrations that leverage the PPR in different kinds of hosts. In comparison to three simple energy-efficient algorithms for allocating and selecting VMs, ThrRs, MadMmt, and lqrMc, the experimental performance showed that the system could reduce energy consumption to 69.31% for many types of host computers, including certain shutdown speeds and migration times along with a minor degradation in performance for cloud data centres [12,13].

Jim et al. (2019) implemented a VM allocation in a cloud data centre with a speed switch and VM consolidation, and the energy efficiency and response results were considered by the data centre. Consequently, to study the queuing model, the authors set up a contiguous multiple-server queuing model [13].

Li et al. (2019) proposed the latest VM allocation approach for power consumption and load balancing to resolve security issues [14].

Nashaat et al. (2019) developed an updated smart elastic scheduling algorithm (SESA) to cluster the VMs to be migrated. SESA aimed to obtain a load-balanced allocation and migration policy. SESA used CPU utilization and RAM as the parameter for the VMs to be located on the same physical machine [8].

Gamal et al. (2019) consider load as a primary parameter for VM allocation and migration. In order to migrate the VMs, a bioinspired VM migration policy was presented, which uses RAM and CPU utilization as the main parameters [15].

Basu S. et al. (2019) have resolved the problem of workload using nature-inspired GA along with the improved scheduling of VMs. Every chromosome of GA has been defined as a node, and the VM is allocated to a node related to the chromosome's genes. After that, based on the crossover and mutation operator, the allocation of VM has been performed. The results have indicated that the proposed approach has performed well in terms of resource utilization and load balancing. However, the area of cloud computing and VM allocation is not new. However, still, the proposed algorithm presents a new behaviour of the cuckoo search algorithm for the selection of VMs to be migrated. The selected VMs are cross-validated using a feed-forward back propagation model which dual checks the allocation, which further ensures optimal power consumption [16].

Zhang et al. (2019) have presented a new, as well as effective, evolutionary-based strategy to allocate VM and thereafter minimized the energy consumption of CDC. The comparative analysis through performing simulation on CloudSim has demonstrated that the proposed method enables for rapidly implementing an allocation corresponding to a set of VMs in an optimized way and integrating more VMs with fewer PMs. Higher energy efficiency than baseline energy efficiency methods has been achieved. Specifically, compared with the

most advanced methods, the complete profit gain as well as saves consumed energy in a preset approach up to 24 % and 41 %, respectively. In addition, our approach can enable CDC to satisfy more end-user requests [17].

• **Motivation of the Research**
 • **Methodology**
 • **Results and Discussion**
 • **Conclusions**
 • **Acknowledgments**
 • **Conflicts of Interest**
 • **References**

na et al. (2019) have presented a Modified PSO method in which the researchers have considered two parameters, namely average scheduling length along with the rate to execute effectively. The comparison with baseline techniques such as Max–Min, Min–Min, and simple PSO approaches have been performed to analyse the effectiveness of the proposed work. The main motive of this paper is to allocate resources by properly managing the workload in a cloud server, which is performed by using a task scheduling algorithm [18].

• **Discussion**
 • **Conclusion**
 • **Acknowledgments**
 • **Conflicts of Interest**
 • **References**

Gawali and Shinde (2018) have used a metaheuristic approach by integrating the two optimization approaches, namely, bandwidth-aware divisible scheduling with BAR optimization for task scheduling along with solving the purposes of resource allocation. Using the modified analytic hierarchy process (MAHP), each task is processed. The experiments have been performed on a cyber snake in which epigenomics is passed as an input task. The performance in terms of turnaround time and response time has been compared to the BATS and IEA approaches. The proposed approach found better results with high resource utility [19].

Verma and Kaushal (2017) have proposed an integrated approach and given the name of “Hybrid Particle Swarm” optimization (HPSO). The algorithm is used to maintain the scheduling workflow problem using a fitness function designed for an HPSO approach and used in an IaaS cloud. The optimization technique has been integrated into the deadline constraint heterogeneous earliest finish time (HEFT) algorithm [20].

Wang et al. (2017) presented an approach on the basis of classifying memory pages into five categories with the aim to minimize total data transferred, by which total migration time has also been reduced. Such categories are inode, cache, anonymous, free memory pages, and kernel, by which they transmit only the first three categories of memory pages since they are required for the execution of the kernel. However, normal execution has not affected by free memory pages, avoiding the transfer of cache memory pages leads to degradation of performance because of inconsistency among actual transmitted memory pages along with the kernel state. The attained findings contrast with the default per copy of KVM and indicated that this scheme can minimize average migration time by 72% [21].

Kansal et al. (2016) proposed a firefly algorithm-based approach in which the maximum loaded VMs are migrated to the least loaded PM without violating the performance of energy efficiency in CDC. From experiments, the improvement in energy consumption of 44.39%, minimization in migration of 72.34%, and hosts of up to 34.36% have been saved [6].

Akbar et al. (2016) introduced a recent task scheduling approach named median deviation-based task scheduling (MDTS) that utilized median absolute deviation (MAD) of the expected time to compute (ETC) in a task as a key attribute to determine ranks of that task. Authors utilized a coefficient-of-variation (COV)-based approach that considered task along with device heterogeneity with the aim to get the ETC of specific DAG. The execution time

cloud program can be visualized as a collection of various tasks depicted through DAG which executes in their logical sequence. To accomplish the improved performance along with enhanced efficiency in a cloud environment, the prioritization of such tasks plays a key role. The evaluation of presented schemes has been done in different circumstances through DAGs with real-world applications. The outcome indicated that the MDTs approach produced high-quality schedulers along with significantly minimising the makespan by 25.01% [22].

Esa and Yousif (2016) introduced a novel job-scheduling mechanism that uses the firefly algorithm with the aim to minimize the time to execute jobs. The presented frameworks for a job along with resource information such as the length of the resource, along with job speed identifiers. In the work-schedule process, the proposed scheduling function first generates a collection of jobs and resources to produce the population by assigning the work to the resources randomly and measuring the population using fitness values that reflect the execution time of work. Next, to provide the shortest execution time for the job, the function consumed iterations to reproduce the population on the basis of the actions of the fireflies to deliver the best job plan. The Java language and the CloudSim simulator are used to implement various scenarios [23].

R. Durga Lakshmi (2016) proposed a minor modification in the working architecture of the genetic algorithm (GA) to optimize the overall system efficiency. Quality of service (QoS) management distributes the problem generated by cloud applications. Tools were addressed to guarantee service levels in terms of performance, reliability, availability and so on. The system's waiting time should be reduced to boost the QoS in the system. GA is a technique of heuristic search that provides the optimal solution for the task. The discussed approach produces a scheduling algorithm based on GA to maximize the system's overall latency. The cloud environment is split into two parts: the cloud user (CU) is one and the cloud-service provider (CSP) is another. A service request is sent to the CSP by the CU and all the requests have been stored in the request. The queue (RQ) explicitly interacts within the C with the GA module queue sequencer (GAQS) [24].

Deshpande et al. (2016) suggested a combining copying technique for live VM migration. The amount of transferred memory pages has decreased by this technique. The hidden pages for dirtied work can be demanded by the destination or actually printed through the source [25].

Forsman et al. (2015) have presented two schemes to cooperate for automated live migration among multiple VMs. The researchers have used a push scheme in order to migrate VMs from overload PMs to under-loaded PMs. Using this strategy, the under-utilized PMs have requested to the heavy-workload servers that the load can be passed to under-utilized servers. Using three conditions, namely, PM state after migration, cost of migration, and workload distribution, at what time VM should be migrated was determined. The redistribution of workload and the quick attainment of a balanced cloud system has been the key objective of this work, whereas, how SLA violation has been minimized was not explained in this paper [26].

Back to Top

M. S. Pilavare et al. (2015) have presented a number of schemes to enhance the existing load-balancing algorithms in CDC. Among all schemes, the researchers have stated that GA is superior to various technologies. GA uses a randomly selected processor as the input and then processes it. It was thought that the same priority was given to processors and jobs but this is not the case. The logarithmic matrix with the least squares is provided here to increase the efficiency of the GA. When a PM is not performing anything, ethically, it is termed as idle, which wastes the available resources and, hence, to solve the problems of idleness through observation, the proposed algorithm performed better. It has been concluded by observing the method of the GA randomly selected processors and concluded that the processors with a higher fitness value have been utilized and that the virtual machines with lower fitness values have left. The simulation has been done using a cloud simulator [27].

Garg et al. (2014) have resolved the problem of resource allocation in CDC in which the task has been performed with different workload conditions using distinct applications. An admission control, along with a scheduling mechanism, has been proposed which can not only maximize the use of resources but also guaranteed that the end-user QoS requirements must be satisfied along with the SLA parameters. From the research, it has been concluded that the understanding of different kinds of SLAs and the applicable penalties along with the workload mix are very important for better resource allocation and utilization of CDC [28].

Song et al. (2014) presented a method of resource allocation as an online box problem. To dynamically assign available resources in the cloud data centre according to the application requirements and maximize the number of active servers to help green computers, virtualization technology has been used. To resolve this problem, a variable item size boxing (VISBP) algorithm has been presented that has been implemented carefully using both VM and PM as classification functions. Small variation has been tolerated as long as the classification rules have been satisfied by the system. The finding indicated that the performance of VISBP is better for migration of a hotspot as well as in load balancing, in contrast to the existing algorithms. The key difficulty in its implementation may be the assumption that all PMs in this work have homogeneous unit power [29].

Beloglazov et al. (2012) investigated the research on energy-saving computing and put forward: (i) cloud energy-saving management architecture principles; (ii) energy-saving resource allocation strategies and scheduling algorithms that consider the QoS expectations and power user features of devices; and (iii) deal with some openness research challenges, such challenges may offer immense advantages for suppliers of resources and customers. By performing a performance assessment using the CloudSim toolkit, the methodology was validated. The findings showed that the model of cloud computing has great potential because it can save a lot of costs and has great potential in complex workload scenarios to increase energy efficiency [1].

Quang-Hung et al. (2013) proposed a genetic algorithm for VM allocation. For experiments, the workload in a computer lab at a university for one day has been examined.

The VMs are sorted in start time using the BFD algorithm. The results obtained using the proposed GAPA scheme have been compared with the existing approach and find that the proposed algorithm achieves less energy consumption. The results show that the completion time has been reduced to a great extent [30].

Madhusudhan et al. (2013) introduced an energy-aware VM placement strategy in combination with swarm-inspired particle swarm optimization (PSO) algorithm to minimize the total energy consumption rate and hence enhances server usage and provide satisfaction to mobile cloud users. The parameters, as well as the operators, have been redefined to resolve the discrete optimization issues. The parameters have been readjusted as the traditional PSO technique has failed to solve the problem of VM placement since it has been mostly utilized for solving only continuous optimization complexities. The PSO algorithm helps to minimize the searching time and, hence, saves energy with better server utilization [31].

Talwar and Singla (2021) suggested using an extended artificial bee colony (E-ABC) method. Results from simulations show that the E-ABC approach has high scalability [32–33]. The E-ABC strategy saved 15–17% more energy and resulted in 10% fewer migrations than the current concept [35]. The proposed approach has a higher computation time.

Dia et al. (2022) studied a cloud-assisted fog computing framework with task offloading and service caching presented to ensure effective task processing. The framework allowed tasks to make offloading decisions for local processing, fog processing, and cloud processing, with the objective of minimizing task delay and energy consumption, taking into account dynamic service caching. To achieve this, a distributed task offloading algorithm based on noncooperative game theory was proposed. Furthermore, the 0–1 knapsack method was employed to realize dynamic service caching. Adjustments were made to the offloading decisions for tasks that were offloaded to the fog server but lacked caching service support [36].

Tran et al. (2022) addressed the challenges related to the migration of machines in cloud-computing architecture. The authors proposed a VM migration algorithm based on the concept of Q-learning and Markov decision-making models. The work comprised a training phase followed by the extraction phase. The superiority of the proposed algorithm in terms of feasibility and strengths of the extraction phase is demonstrated using comparative analysis against a max–min ant system, round robin and ant system algorithms [37].

Abedi et al. (2022) considered the heuristic method using the firefly algorithm and fuzzy approach to prioritize the tasks. The authors improved the population of the firefly algorithm with the intention of balancing the load, migration rate, average run time, and completion time. The authors used MATLAB software for implementation and to maintain the makespan to 50.16. The drawback of the study was that some objectives still need to achieve using better optimization [38].

Khan et al. (2022) presented a hybrid cuckoo search and particle swarm optimization (CU-PSO) approach for effective VM migration. The primary goals of this study are to

the duration of computations, decrease energy use, and lower the cost of relocation. The optimal use of available resources is another focus area. The study aim is supported by a simulation analysis that compares the efficiency of the hybrid optimization model to that of more traditional methods. The proposed approach is outperformed in terms of energy consumption, migration cost, resource availability, and computation time [39]. The proposed work is insufficient to address the issue of SLA violation.

Motivation of the Research

Methodology

Results and Discussion

Zhao, H. et al. (2023) suggested a VM performance-aware approach (PAVMM) using ant-colony optimization (ACO). The process continues by setting a goal of user-friendly VM performance optimization. The suggested work has goals for cloud-service vendors including reducing the overall migration cost and the number of active PMs. The proposed framework minimizes the energy consumption and minimizes the active host which improves the speed of operation but proposed work lacking in SLA violations [40].

Conclusions

Major Contributions

Discussion

Institutional Review Board

Statement of Ethics

Conflicts of Interest

References

Bail, et al. (2023) developed a priority-aware task scheduling (PATS) algorithm specifically designed for sensor networks. The algorithm aims to schedule priority-aware tasks for data offloading on edge and cloud servers. The proposed sensor-network design includes dividing tasks into four distinct groups: very urgent (OVU), urgent (OU), moderate (OM), and nonurgent (ONU). To address this problem, a multiobjective function formulation was used and the efficiency of the algorithm was evaluated using the bio-inspired NSGA-2 technique. The results obtained demonstrated significant improvement compared to benchmark algorithms, highlighting the effectiveness of the proposed solution. Additionally, when the number of tasks was increased from 200 to 1000, subsequent improvements were observed. Specifically, the average queue delay, computation time, and energy showed overall improvements of 17.2%, 7.08%, and 11.4% respectively, for the 200-task scenario [41].

Singh et al. (2023) focused on the utilization of containerized environments, specifically docker, for big data applications with load balancing. A novel scheduling mechanism for containers in big-data applications was proposed, based on the docker swarm and the microservice architecture. Docker swarm was employed to effectively manage the workload and service discovery of big-data applications. The implementation of the proposed mechanism involved a case study that was initially deployed on a single server and then scaled to four instances. The master node, implemented using the NGINX service, ran the swarm commands, while the other three services were microservices dedicated to the implemented scenario for a big-data application. These three worker nodes consisted of the PHP front-end microservice, Python API microservice, and Redis cache microservice. The results of the study demonstrated that increasing workloads in big-data applications could be effectively managed by utilizing microservices in containerized environments, and docker swarm enabled efficient load balancing. Furthermore, applications developed using containerized microservices exhibited reduced average deployment time and improved continuous integration [42].

- Kayitha et al. (2023) developed a novel approach called filter-based ensemble feature selection (FEFS) combined with a deep-learning model (DLM) for intrusion detection in cloud computing. The DLM utilized a recurrent neural network (RNN) and Tasmanian devil optimization (TDO) to determine the optimal weighting parameter. The initial phase involved collecting intrusion data from global datasets, namely KDDCup-99 and NSL-KDD, which were used for validating the proposed methodology. The collected database was then utilized for feature selection to enhance intrusion prediction. FEFS, a combination of filter, wrapper, and embedded algorithms, was employed for feature extraction, selecting essential features for the training process in the DLM. The proposed technique was implemented using MATLAB and its effectiveness was evaluated using performance metrics such as sensitivity, F-measure, precision, recall, and accuracy. The suggested strategy demonstrated significant improvements based on these performance metrics. A comparison was made between the proposed method and conventional techniques such as RNN, deep neural network (DNN), and RNN-genetic algorithm (RNN-GA) [43].

3. Motivation of the Research

This research work is inspired by the research done by Beloglazov et al. [1], Kansal et al. [6], and Nashaat et al. [8]. There is a possibility of enhancement in the SESA algorithm by adding a third variable, bandwidth utilization (BU), as done in [3]. In addition, the SESA can be improved by adding similarity measures other than ED. In addition to this, the allocation is done based on the clusters' density only, which could be further improved.

4. Methodology

The methodology of the proposed algorithm is centric, as per the written objectives. The first objective is to analyse existing VM allocation and migration techniques in which the focus would be towards studying SESA, MBFD, and its other enhancement. The design of the proposed algorithm is divided into two segments, namely the organization of the clusters and the placement of VMs into the concerned clusters. Considering the VMs to be migrated, as per the MM algorithm, Nashaat et al. [8] in 2019 proposed a smart elastic scheduling algorithm (SESA) which clusters the VMs in order to migrate. Nashaat modified the k-means algorithm by using RAM and CPU utilization of the VMs and the host. In order to modify the k-means algorithm, Nashaat used Euclidean distance (ED).

4.1. Detection of the Hotspot

Detection of the hotspot, based on the unit threshold, which may be SLAV or load, is a common practice. Even if a PM is idle, it consumes 70% of its peak power and, hence, letting the PM sit idle would also be using the power without producing an output. A dual threshold policy which keeps one upper threshold and one lower threshold of utilization is observed to be practised commonly [11,12,13]. The upper and the lower threshold are responsible for the

determination of the fact that whether all the VMs should be migrated from the PM or some VMs should be migrated.

[Introduction](#)

[Order Article Reprints \(/1424-8220/23/13/6117/rer](#)

4.2. Selection of VMs

[toggle desktop layout cookie](#)  

- Motivation of the Research
 - Methodology
 - Results and Discussion
 - Conclusions
- utilization, time of migration, and correlation between the VMs [44,45]. Considering the essentials to reduce the power consumption, minimization of migrations (MM) and minimum utilization (MU) were proposed and have gained potential enhancements time by time [6,14].

4.3. Selection of PM

- Acknowledgments
 - Funding
 - Institutional Review Board
 - Conflicts of Interest
 - References
- as per the required essentials and then the VMs are placed as per shown in **Section 4.4**. The listed PMs are now called the available PMs.

4.4. Data Migration Statement

- Acknowledgments
 - Conflicts of Interest
 - References
- Putting the right VM to the right PM is the task of this section. The proposed CESSA algorithm enhances the k-means algorithm by using CPU utilization (CPU), bandwidth utilization (BU), and currently associated random access memory (AsRAM). The algorithm takes inspiration from Nashaat et al.'s clustering algorithm, the smart elastic scheduling algorithm (SESA) [8]. CESSA is subdivided into three parts for the calculation of the number of centroids and the identification of centroids, placement of VMs into the centroid, and prioritization of the created clusters.

The resource assignment or allocation [46,47,48] has four steps for VM allocation, as demonstrated in **Figure 2**. The first step is the detection of the hotspot PM. The SESA algorithm given in Algorithm 3 finds the optimal number of clusters required for the VMs to be migrated. In order to excel in the performance, an association of VM to PM is done using the Euclidian distance.

Algorithm 3: SESA

Input: hostList, VMList, Standard Deviation threshold, Output: high density arranged cluster list of co-located VMs, allocation of VMs

1. Find K points for selecting the optimal number of clusters
2. Calculating K1 (for Parameter – > CPU)
3. $K1\_maxpoint = hostlist.get\_max(CPU)/VMList.get\_min(CPU)$
4. $K1\_minpoint = hostlist.get\_min(CPU)/VMList.get\_max(CPU)$
5. $K1 = average(K1\_maxpoint, K1\_minpoint)$
6. Calculating K2 (for Parameter – > RAM)
7. $K2\_maxpoint = hostlist.get\_max(RAM)/VMList.get\_min(RAM)$
8. $K2\_minpoint = hostlist.get\_min(RAM)/VMList.get\_max(RAM)$
9. $K2 = average(K2\_maxpoint, K2\_minpoint)$
10. $K = average(K1, K2)$

[Back to Top](#)

- 11. Select the initial centroid as a pair of two values (CPU, RAM)

Download PDF (/1424-8220/23/13/6117/pdf?version=1688376187)

- 12. $\text{centroids}[1, 1] = \text{Get\_average\_CPU}(\text{VMList})$

- 13. $\text{centroids}[1, 2] = \text{Get\_average\_RAM}(\text{VMList}) *$

- 14. Find the remaining $K - 1$ centroids,

- 15. for each mth centroid number do, Where m takes values from 1 to $K - 1$

- 16. Calculate the Euclidian distance (ED) between previous centroid and (CPU, RAM)

- 17. Calculate the Euclidian distance (ED) between previous centroid and (CPU, RAM)

- 18. for each VM in VMList do,

- 19. where j takes values from 1 to no. of VMs in VMList

- 20. $\text{Euclidian}(\text{VMList.get}(j), \text{centroids}[m, 1], \text{centroids}[m, 2]) *$

- 21. $\text{ED}[m] = \text{find\_Euclidian}(\text{VMList.get}(j), \text{centroids}[m, 1], \text{centroids}[m, 2]) *$

- 22. end for

- 23. $\text{centroids}[m + 1, 1] = \text{VMList.get}(\text{get\_index\_forMaxValue}(\text{Ecu\_dis})).\text{get}(\text{CPU})$

- 24. $\text{centroids}[m + 1, 2] = \text{VMList.get}(\text{get\_index\_forMaxValue}(\text{Ecu\_dis})).\text{get}(\text{RAM})$

- 25. end for

- 26. Calculate the ED between each VMs and all Cluster's centroids

- 27. for each jthVM in VMList do, Where j takes values from 1 to no. of VMs in VMList

- 28. for each mth centroid number do, Where mtakes values from 1 to $K - 1$

- 29. $\text{ED}[m] = \text{find\_Euclidian}(\text{VMList.get}(j), \text{centroids}[m, 1], \text{centroids}[m, 2]) *$

- 30. end for

- 31. Append VM to the Cluster with minimumED

- 32. $\text{Cluster} = \text{Append\_in\_Cluster}(\text{get\_index\_forMinValue}(\text{ED}), \text{VMList.get}(j))$

- 33. end for

- 34. Arrange the co-located VMs

- 35. for each ith VMs cluster list in cluster do

- 36. $\text{arrangeBy Co-locatedVMs}(\text{Cluster.get}(i))$

- 37. end for

- 38. $\text{VMList} = \text{arrangeBy HighDensityCluster}(\text{Cluster})$

Order Article Reprints (/1424-8220/23/13/6117/reprints)

toggle desktop layout cookie

- Download PDF (/1424-8220/23/13/6117/pdf?version=1688376187)
- Introduction
- Literature Survey
- Motivation of the Res
- Methodology
- Results and Discussi
- Conclusions
- Author Contributions
- Funding
- SciProfiles
- Institutional Review B
- Statement
- groups/public/10.3
- Informed Consent Sta
- Data Availability State
- Acknowledgments
- Conflicts of Interest
- References

Comment

4.5. VM Migaion

PSEUDO-CODE CESCA

Inputs: H_{List} , VM_{List} Output: prioritized

//Calculate the total number of centroids P

//Calculate P_x , P_y and P_z , Where P_x is based on CPU, P_y is calculated based on AsRAM

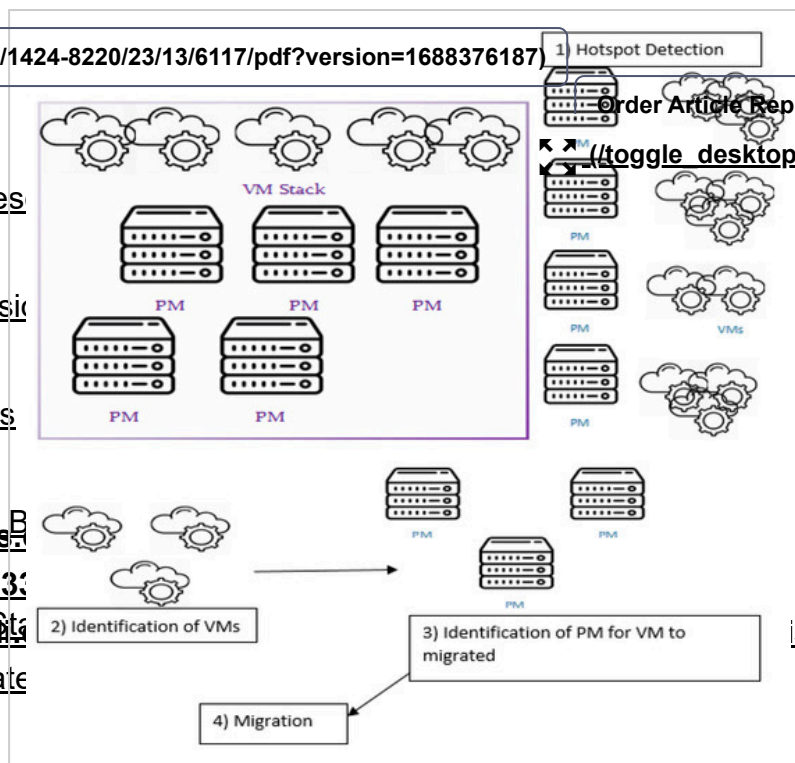


Figure 2. VM Allocation [1,8].

- [Download PDF \(1424-8220/23/13/6117/pdf?version=1688376187\)](#)

[Introduction](#) [Share](#) [Literature Survey](#) [Motivation of the Research](#) [Methodology](#) [Results and Discussion](#) [Conclusions](#) [Author Contributions](#) [Funding](#) [Institutional Review Board Statement](#) [Informed Consent Statement](#) [Data Availability Statement](#) [Acknowledgments](#) [Conflicts of Interest](#) [References](#)

$$P_{x_{max}} = \frac{H_{List} \cdot get_{max}(CPU)}{VM_{List} \cdot get_{min}(CPU)}; P_{x_{min}} = \frac{H_{List} \cdot get_{min}(CPU)}{VM_{List} \cdot get_{max}(CPU)}$$

$$P_y = \frac{P_{x_{min}} + P_{x_{max}}}{2}$$

Motivation of the Research

$$P_{y_{max}} = \frac{H_{List} \cdot get_{max}(AsRAM)}{VM_{List} \cdot get_{min}(AsRAM)}; P_{y_{min}} = \frac{H_{List} \cdot get_{min}(AsRAM)}{VM_{List} \cdot get_{max}(AsRAM)}$$

Results and Discussion

Conclusions

$$P_{z_{min}} = \frac{H_{List} \cdot get_{min}(BU)}{VM_{List} \cdot get_{max}(BU)}; P_{z_{max}} = \frac{H_{List} \cdot get_{max}(BU)}{VM_{List} \cdot get_{min}(BU)}$$

$$P_z = \frac{P_{z_{min}} + P_{z_{max}}}{2}$$

[https://sciprofiles.com/discussion-growth/public/10.3390/s23136117?utm_source=twitter&utm_medium=publication&utm_campaign=discuss_in_sciprofiles\)](#)

Informed Consent Statement

[Data Availability Statement](#) [Acknowledgments](#) [Conflicts of Interest](#) [References](#)

[Data Availability Statement](#) [Acknowledgments](#) [Conflicts of Interest](#) [References](#)

[Data Availability Statement](#) [Acknowledgments](#) [Conflicts of Interest](#) [References](#)

[Data Availability Statement](#) [Acknowledgments](#) [Conflicts of Interest](#) [References](#)

[Data Availability Statement](#) [Acknowledgments](#) [Conflicts of Interest](#) [References](#)

[Data Availability Statement](#) [Acknowledgments](#) [Conflicts of Interest](#) [References](#)

[Data Availability Statement](#) [Acknowledgments](#) [Conflicts of Interest](#) [References](#)

[Comment](#)

$$Cos_{sim} = \frac{\sum_{i=1}^n h1_i \times h2_i}{\sqrt{\sum_{k=1}^n h1_k^2} \times \sqrt{\sum_{k=1}^n h2_k^2}} \quad (11)$$

i Calculate the rest (P-1) of the centroids

7. $VM_{remaining} = VM_{List}$
8. $CS = []$; // Initialize cosine similarity to 0
9. if $P > 0$
10. $CS[i] = GetCoSim(VM_{remaining}[i].getattributes(), 1st\ centroid)$
11. $CS_{index} = Findmin_{index}(CS)$
12. $Next_{Centroid}(1,1) = VM_{List}.CS_{index}.get(CPU)$
13. $Next_{Centroid}(1,1) = VM_{List}.CS_{index}.get(RAM)$
14. $Next_{Centroid}(1,1) = VM_{List}.CS_{index}.get(BU)$
15. $P = P - 1$
16. $VM_{remaining}.Remove[CS_{index}]$
17. End if
18. End for
19. For each vm in $VM_{remaining}$
20. $Simv = []$ // Initialize simv to empty
21. For each $cent$ in $Centroid$
22. $Simv.append(GetCoSim(VM, cent))$
23. End For
24. $Find\ min\ v = min_{index}(Simv)$
25. Allocate vm to $Centroid.min\ v_{index}$

// Prioritization of the created clusters

1. $avg_{similarity} = []$
2. Calculate $avg_{similarity}$ and Append to $avg_{similarity}$
3. Append $avg_{similarity}$ to $avg_{similarity}$
4. End for
5. $Prioritized = Sort.ascending(avg_{similarity})$
6. Return $Prioritized$

Back to Top

- The proposed work modifies the current state of art with the alterations in the calculation of the similarity indexes. With the alteration in the similarity evaluation and added parameter of BU, the proposed work results in better power efficiency and violation ratios. The results are illustrated in the next section.

Motivation of the Research

- Methodology.

5. Results and Discussion

Results and Discussion

- Conclusions to compare the proposed work with existing algorithm architectures, two parameters, namely the overall consumed power and SLA-V, and in order to compute the SLA-V, the power consumption has been utilized along with a total number of migrations performed. The power consumption has been evaluated in watts whereas the SLA-V is unit less and is stated in Table 1.
- Informal Consent Statement
- Data Availability Statement
- Acknowledgments
- Conflicts of Interest
- References

The analysis has been done taking the total VM in the incremental ratio of 100 VMs per 10 PMs, viz. the VMs range is 100–1000 VMs whereas the PMs range is 10–100 PMs. The power consumption is in watts and has been evaluated using Equation (3). As the total number of VMs increases, the average consumption also increases gradually. It is not necessary that if the load increases and the power consumption should also increase until and unless the variation of the load is significantly high. In most of the observed cases, the average power consumption increases. The average consumption of power can be illustrated using **Figure 3** as follows.

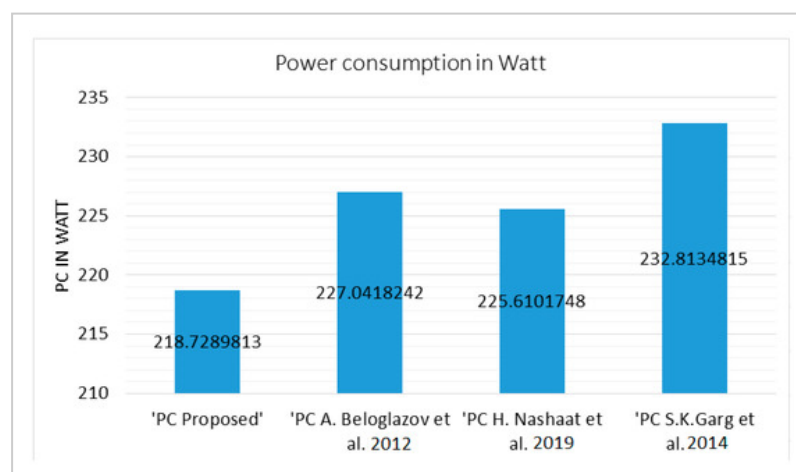


Figure 3. Average Power Consumption against [1,8,28].

• In order to evaluate one scenario, the proposed work and other compared algorithms/techniques have been evaluated for 10,000 simulations. The average consumed power in the case of the proposed work scenario is improved by 3.8% as compared to [1], 3.14% as compared to [8], and 3.92% as compared to [28]. The effect of migration is evident when the total number of VMs increases after 200 VMs. When the load increases over the system, the system with sustainable allocation architecture will be more efficient in terms of holding the VMs at the right PM and discarding the VMs that are not required at the moment.

As shown in **Figure 3**, the proposed work demonstrates minimum power consumption with the least number of migrations, the performance is evaluated to be improved overall. This is due to the additional evaluation measure added to the distance calculation and with additional input parameters, viz. DC in the assembly of the VMs. The proposed work evaluates the overall SLA-V based on both the parameters, including the total number of migrations and total power consumption, as follows.

As the proposed work balances both the total number of migrations and the power consumption parallelly, the overall SLA-V for the proposed work is quite economical in relation to other state-of-art techniques, as shown in **Figure 4**. The average overall SLA-V for the proposed work is 0.201 in all the scenarios, whereas [1,8,28] demonstrate 0.241, 0.2314, and 0.2712 for the same load conditions.

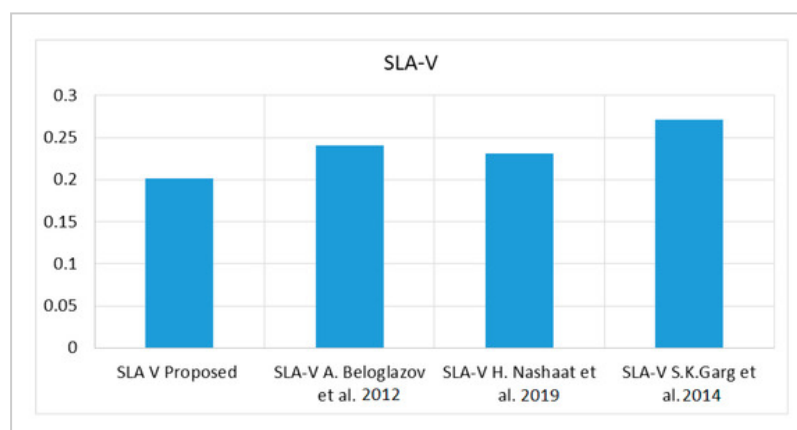


Figure 4. SLA-V against [1,8,28].

6. Conclusions

This paper presented an advanced algorithm architecture that was inspired by Nashaat et al. [8] architecture. The proposed work modified the current state of art technique by adding two utility parameters to the system, namely the bandwidth utilization and the cosine similarity to enhance the overall consumed power in order to allocate and migrate the VMs from the Nashaat et al. presented SESA algorithm, which is enhanced by utilizing more similarity indexes [8]. This research work is inspired by the research done by Beloglazov et al. [1], Kansal et al. [6], Nashaat et al. [8] and Garg et al. [28]. Due to improved utility function, the proposed work outcasts [1,8,28] by 3.8% as compared to [1], 3.14% as compared to [8],

Back to Top

and 3.92% as compared to [28]. The evaluation has also been made on the basis of the total number of migrations and as the VM selection is precise in the case of the proposed work; the VMs are migrated in less quantity maintaining the overall power consumption in the case of the proposed work. The overall SLA V in all the presented scenarios, viz. 10–100 PMs supplied for 100–1000 VMs, is the least, as compared to other state-of-art techniques.

Results and Discussion

Author Contributions

All the following authors contributed to this research work: A.K. (Amandeep Kaur), S.K., D. G., Y.H., M.H., A.K. (Amel Ksibi), H.E. and S.S.; Conceptualization, A.K. (Amandeep Kaur) and D.G.; software, S.K. and Y.H.; validation, D.G. and S.S.; formal analysis, A.K. (Amandeep Kaur) and S.K.; investigation, S.S.; resources, M.H. and A.K. (Amel Ksibi); information resources, A.K. (Amandeep Kaur) and A.K. (Amel Ksibi); writing—original draft preparation, A.K. (Amandeep Kaur) and S.K.; writing—review and editing, D.G. and S.S.; visualization, Y.H. and D.G.; supervision, D.G.; project administration, H.E., A.K. (Amel Ksibi) and M.H.; funding acquisition, H.E., A.K. (Amel Ksibi) and M.H. All authors have read and agreed to the published version of the manuscript.

Funding

This work is supported by the Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2023R125), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Data Availability Statement

Data would be made available on request.

Acknowledgments

The authors would like to thank the Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2023R125), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Conflicts of Interest

- The authors declare no conflict of interest.

[Order Article Reprints \(/1424-8220/23/13/6117/reprints\)](#)

- Literature Survey.

[\(/toggle desktop layout cookie\)](#)

References

1. Beloglazov, A.; Abawajy, J.; Buyya, R. Energy-Aware Resource Allocation Heuristics for Efficient Management of Data Centers for Cloud Computing. *Future Gener. Comput. Syst.* **2012**, *28*, 755–768. [Google Scholar] (https://scholar.google.com/scholar_lookup?title=Energy-Aware+Resource+Allocation+Heuristics+for+Efficient+Management+of+Data+Centers+for+Cloud+Computing&author=Beloglazov,+A.&author=Abawajy,+J.&author=Buyya,+R.&publication_year=2012&journal=Future+Gener.+Comput.+Syst.&volume=28&pages=755%E2%80%93768&doi=10.1016/j.future.2011.04.017) [CrossRef (<https://doi.org/10.1016/j.future.2011.04.017>)] [Green Version (<http://beloglazov.info/papers/2011-energy-aware-fgcs.pdf>)]
2. Baker, B.S.; Coffman, E.G., Jr. A Tight Asymptotic Bound for Next-Fit-Decreasing Bin-Packing. *SIAM J. Algebr. Discret. Methods* **1981**, *2*, 147–152. [Google Scholar] (https://scholar.google.com/scholar_lookup?title=A+Tight+Asymptotic+Bound+for+Next-Fit-Decreasing+Bin-Packing&author=Baker,+B.S.&author=Coffman,+E.G.,+Jr.&publication_year=1981&journal=SIAM+J.+Algebr.+Discret.+Methods&volume=2&pages=147%E2%80%93152&doi=10.1137/0602019) [CrossRef (<https://doi.org/10.1137/0602019>)]
3. Lu, X.; Zhang, Z. A Virtual Machine Dynamic Migration Scheduling Model Based on MBFD Algorithm. *Int. J. Comput. Theory Eng.* **2015**, *7*, 278. [Google Scholar] (https://scholar.google.com/scholar_lookup?title=A+Virtual+Machine+Dynamic+Migration+Scheduling+Model+Based+on+MBFD+Algorithm&author=Lu,+X.&author=Zhang,+Z.&publication_year=2015&journal=Int.+J.+Comput.+Theory+Eng.&volume=7&pages=278)
4. Mann, Z.Á. Multicore-Aware Virtual Machine Placement in Cloud Data Centers. *IEEE Trans. Comput.* **2016**, *65*, 3357–3369. [Google Scholar] (https://scholar.google.com/scholar_lookup?title=Multicore-Aware+Virtual+Machine+Placement+in+Cloud+Data+Centers&author=Mann,+Z.&publication_year=2016&journal=IEEE+Trans.+Comput.&volume=65&pages=3357%E2%80%933369&doi=10.1109/TC.2016.2529629) [CrossRef (<https://doi.org/10.1109/TC.2016.2529629>)]

- [Download PDF](#) (/1424-8220/23/13/6117/pdf?version=1688376187)
 - [Introduction](#)
 - [Share](#)
 - [Literature Survey](#)
 - [L.N. & publication_year=2007&journal=Phys.+Life+Rev.&volume=4&pages=1%E2%80%9336&doi=10.1016/j.plrev.2006.10.002\)\]](#) [CrossRef] (
 - [Methodology](#)
 - [Results and Discussion](#)
 - [Conclusions](#)
 - [Cite this article](#)
 - [Kansal, N.J.; Chana, I. Energy-Aware Virtual Machine Migration for Cloud Computing-a Fuzzy-Based Optimization Approach. J. Grid Comput. 2016, 14, 327–345. \[Google Scholar\] \(\[https://scholar.google.com/scholar_lookup?title=Energy-Aware+Virtual+Machine+Migration+for+Cloud+Computing-Statement+of+Optimization+Approach&author=Kansal,+N.J.+&author=Chana,+I.+&publication_year=2016&journal=J.+Grid+Comput.&volume=14&pages=327%E2%80%93345&doi=10.3390/s23136117\]\(https://scholar.google.com/scholar_lookup?title=Energy-Aware+Virtual+Machine+Migration+for+Cloud+Computing-Statement+of+Optimization+Approach&author=Kansal,+N.J.+&author=Chana,+I.+&publication_year=2016&journal=J.+Grid+Comput.&volume=14&pages=327%E2%80%93345&doi=10.3390/s23136117\)\)\]](#) [CrossRef] (
 - [Data Availability Statement](#)
 - [Acknowledgments](#)
 - [Conflicts of Interest](#)
 - [References](#)
 - [Comment](#)
8. Nashaat, H.; Ashry, N.; Rizk, R. Smart Elastic Scheduling Algorithm for Virtual Machine Migration in Cloud Computing. *J. Supercomput.* **2019**, *75*, 3842–3865. [Google Scholar] ([\[CrossRef\] \(](https://scholar.google.com/scholar_lookup?title=Smart+Elastic+Scheduling+Algorithm+for+Virtual+Machine+Migration+in+Cloud+Computing&author=Nashaat,+H.+&author=Ashry,+N.+&author=Rizk,+R.+&publication_year=2019&journal=J.+Supercomput.&volume=75&pages=3842%E2%80%933865&doi=10.1007/s11227-019-02748-2))
9. Masdari, M.; Khezri, H. Efficient VM Migrations Using Forecasting Techniques in Cloud Computing: A Comprehensive Review. *Clust. Comput.* **2020**, *23*, 2629–2658. [Google Scholar] ([\[CrossRef\] \(](https://scholar.google.com/scholar_lookup?title=Efficient+VM+Migrations+Using+Forecasting+Techniques+in+Cloud+Computing:+A+Comprehensive+Review&author=Masdari,+M.+&author=Khezri,+H.+&publication_year=2020&journal=Clust.+Comput.&volume=23&pages=2629%E2%80%932658&doi=10.1007/s10586-019-03032-x))

-  Dubey, K.; Sharma, S.C. An Extended Intelligent Water Drop Approach for Efficient VM Allocation in Secure Cloud Computing Framework. *J. King Saud Univ.-Comput. Inf. Sci.* **2020**, *34*, 3948–3958. [Google Scholar] (https://scholar.google.com/scholar_lookup?title=An+Extended+Intelligent+Water+Drop+Approach+for+Efficient+VM+Allocation+in+Secure+Cloud+Computing+Framework&author=Dubey,+K.&author=Sharma,+S.C.&publication_year=2020&journal=J.+King+Saud+Univ.-Comput.+Inf.+Sci.&volume=34&pages=3948%E2%80%933958&doi=10.1016/j.jksuci.2020.11.001) [CrossRef (<https://doi.org/10.1016/j.jksuci.2020.11.001>)]
-  Joshi, A.S.; Munisamy, S.D. Dynamic Degree Balanced with CPU Based VM Allocation Strategy for Load Balancing. *Inf. Optim. Sci.* **2020**, *41*, 543–553. [Google Scholar] (https://scholar.google.com/scholar_lookup?title=Dynamic+Degree+Balanced+with+CPU+Based+VM+Allocation+Policy+for+Load+Balancing&author=Joshi,+A.S.&author=Munisamy,+S.D.&publication_year=2020&journal=J.+Inf.+Optim.+Sci.&volume=41&pages=543%E2%80%93553&doi=10.1080/02522667.2020.1724618) [CrossRef (<https://doi.org/10.1080/02522667.2020.1724618>)]

12. Ruan, X.; Chen, H.; Tian, Y.; Yin, S. Virtual Machine Allocation and Migration Based on Performance-to-Power Ratio in Energy-Efficient Clouds. *Future Gener. Comput. Syst.* **2019**, *100*, 380–394. [Google Scholar] (https://scholar.google.com/scholar_lookup?title=Virtual+Machine+Allocation+and+Migration+Based+on+Performance-to-Power+Ratio+in+Energy-Efficient+Clouds&author=Ruan,+X.&author=Chen,+H.&author=Tian,+Y.&author=Yin,+S.&publication_year=2019&journal=Future+Gener.+Comput.+Syst.&volume=100&pages=380%E2%80%93394&doi=10.1016/j.future.2019.05.036) [CrossRef (<https://doi.org/10.1016/j.future.2019.05.036>)]
13. Jin, S.; Qie, X.; Hao, S. Virtual Machine Allocation Strategy in Energy-Efficient Cloud Data Centres. *Int. J. Commun. Netw. Distrib. Syst.* **2019**, *22*, 181–195. [Google Scholar] (https://scholar.google.com/scholar_lookup?title=Virtual+Machine+Allocation+Strategy+in+Energy-Efficient+Cloud+Data+Centres&author=Jin,+S.&author=Qie,+X.&author=Hao,+S.&publication_year=2019&journal=Int.+J.+Commun.+Netw.+Distrib.+Syst.&volume=22&pages=181%E2%80%93195&doi=10.1504/IJCND.2019.097654) [CrossRef (<https://doi.org/10.1504/IJCND.2019.097654>)]

- Jia, H.; Liu, X.; Di, X.; Qi, H.; Cong, L.; Li, J.; Yang, H. Security Strategy for Virtual Machine Allocation in Cloud Computing. *Procedia Comput. Sci.* **2019**, *147*, 140–144. [Google Scholar (https://scholar.google.com/scholar_lookup?title=Security+Strategy+for+Virtual+Machine+Allocation+in+Cloud+Computing&author=Jia,+H.&author=Liu,+X.&author=Di,+X.&author=Qi,+H.&author=Cong,+L.&author=Li,+J.&author=Yang,+H.&publication_year=2019&journal=Procedia+Comput.+Sci.&volume=147&pages=140%E2%80%93144&doi=10.1016/j.procs.2019.01.204)] [CrossRef (https://doi.org/10.1016/j.procs.2019.01.204)]
- El-Gamal, M.; Rizk, R.; Mahdi, H.; Elnaghi, B.E. Osmotic Bio-Inspired Load Balancing Algorithm for Virtual Machine Migration in Cloud Computing. *IEEE Access* **2019**, *7*, 42735–42744. [Google Scholar (https://scholar.google.com/scholar_lookup?title=Osmotic+Bio-Inspired+Load+Balancing+Algorithm+in+Cloud+Computing&author=El-Gamal,+M.&author=Rizk,+R.&author=Mahdi,+H.&author=Elnaghi,+B.E.&publication_year=2019&journal=IEEE+Access&volume=7&pages=42735%E2%80%9342744&doi=10.1109/ACCESS.2019.2907615)] [CrossRef (https://doi.org/10.1109/ACCESS.2019.2907615)]

16. Islam, M.; Razzaque, A.; Islam, J. A Genetic Algorithm for Virtual Machine Migration in Heterogeneous Mobile Cloud Computing. In Proceedings of the 2016 International Conference on Networking Systems and Security (NSysS), Dhaka, Bangladesh, 7–9 January 2016; pp. 1–6. [Google Scholar (https://scholar.google.com/scholar_lookup?title=A+Genetic+Algorithm+for+Virtual+Machine+Migration+in+Heterogeneous+Mobile+Cloud+Computing&conference=Proceedings+of+the+2016+International+Conference+on+Networking+Systems+and+Security+(NSysS)&author=Islam,+M.&author=Razzaque,+A.&author=Islam,+J.&publication_year=2016&pages=1%E2%80%936)]
17. Zhang, X.; Wu, T.; Chen, M.; Wei, T.; Zhou, J.; Hu, S.; Buyya, R. Energy-Aware Virtual Machine Allocation for Cloud with Resource Reservation. *J. Syst. Softw.* **2019**, *147*, 147–161. [Google Scholar (https://scholar.google.com/scholar_lookup?title=Energy-Aware+Virtual+Machine+Allocation+for+Cloud+with+Resource+Reservation&author=Zhang,+X.&author=Wu,+T.&author=Chen,+M.&author=Wei,+T.&author=Zhou,+J.&author=Hu,+S.&author=Buyya,+R.&publication_year=2019&journal=J.+Syst.+Softw.&volume=147&pages=147%E2%80%93161&doi=10.1016/j.jss.2018.09.084)] [CrossRef (https://doi.org/10.1016/j.jss.2018.09.084)]

- Jana, B.; Chakraborty, M.; Mandal, T. A Task Scheduling Technique Based on Particle Swarm Optimization Algorithm in Cloud Environment. In *Soft Computing: Theories and Applications*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 525–536. [Google Scholar] (https://scholar.google.com/scholar_lookup?title=A+Task+Scheduling+Technique+Based+on+Particle+Swarm+Optimization+Algorithm+in+Cloud+Environment&author=Jana,+B.&author=Chakraborty,+M.&author=Mandal,+T.&publication_year=2019&pages=525%E2%80%93536)]
- Gawali, M.B.; Shinde, S.K. Task Scheduling and Resource Allocation in Cloud Computing Using a Heuristic Approach. *J. Cloud Comput.* **2018**, *7*, 4. [Google Scholar] (https://scholar.google.com/scholar_lookup?title=Task+Scheduling+and+Resource+Allocation+in+Cloud+Computing+Using+Heuristic+Approach&author=Gawali,+M.B.&author=Shinde,+S.K.&publication_year=2018&journal=J.+Cloud+Comput.&volume=7&pages=4&doi=10.1186/s13677-018-0105-8)] [CrossRef (https://doi.org/10.1186/s13677-018-0105-8)]
- Verma, A.; Kaushal, S. A Hybrid Multi-Objective Particle Swarm Optimization for Scientific Workflow Scheduling. *Parallel Comput.* **2017**, *62*, 1–19. [Google Scholar] (https://scholar.google.com/scholar_lookup?title=A+Hybrid+Multi-Objective+Particle+Swarm+Optimization+for+Scientific+Workflow+Scheduling&author=Verma,+A.&author=Kaushal,+S.&publication_year=2017&journal=Parallel+Comput.&volume=62&pages=1%E2%80%9319&doi=10.1016/j.parco.2017.01.002)] [CrossRef (https://doi.org/10.1016/j.parco.2017.01.002)]
- 21. Wang, C.; Hao, Z.; Cui, L.; Zhang, X.; Yun, X. Introspection-Based Memory Pruning for Live VM Migration. *Int. J. Parallel Program.* **2017**, *45*, 1298–1309. [Google Scholar] (https://scholar.google.com/scholar_lookup?title=Introspection-Based+Memory+Pruning+for+Live+VM+Migration&author=Wang,+C.&author=Hao,+Z.&author=Cui,+L.&author=Zhang,+X.&author=Yun,+X.&publication_year=2017&journal=Int.+J.+Parallel+Program.&volume=45&pages=1298%E2%80%931309&doi=10.1007/s10766-016-0471-0)] [CrossRef (https://doi.org/10.1007/s10766-016-0471-0)]

- Back to TopTop

-  Fersman, M.; Glad, A.; Lundberg, L.; Ilie, D. Algorithms for Automated Live Migration of Virtual Machines. *J. Syst. Softw.* **2015**, *101*, 110–126. [Google Scholar (https://scholar.google.com/scholar_lookup?title=Algorithms+for+Automated+Live+Migration+of+Virtual+Machines&author=Fersman,+M.&author=Glad,+A.&author=Lundberg,+L.&author=Ilie,+D.&publication_year=2015&journal=J.+Syst.+Softw.&volume=101&pages=110%E2%80%93126&doi=10.1016/j.jss.2014.11.044)] [CrossRef (https://doi.org/10.1016/j.jss.2014.11.044)] [Green Version (http://bth.diva-portal.org/smash/get/diva2:833828/FULLTEXT02)]
-  Pilavare, M.S.; Desai, A. A Novel Approach towards Improving Performance of Load Balancing Using Genetic Algorithm in Cloud Computing. In Proceedings of the 2015 International Conference on Innovations in Information, Embedded, and Communication Systems (ICIIECS), Coimbatore, India, 19–20 March 2015; pp. 1–4. [Google Scholar (https://scholar.google.com/scholar_lookup?title=A+Novel+Approach+towards+Improving+Performance+of+Load+Balancing+Using+Genetic+Algorithm+in+Cloud+Computing&conference=Proceedings+of+the+2015+International+Conference+on+Innovations+in+Information,+Embedded+and+Communication+Systems+(ICIIECS)&author=Pilavare,+M.S.&author=Desai,+A.&publication_year=2015&pages=1%E2%80%934)]
- 28. Garg, S.K.; Toosi, A.N.; Gopalaiyengar, S.K.; Buyya, R. SLA-Based Virtual Machine Management for Heterogeneous Workloads in a Cloud Datacenter. *J. Netw. Comput. Appl.* **2014**, *45*, 108–120. [Google Scholar (https://scholar.google.com/scholar_lookup?title=SLA-Based+Virtual+Machine+Management+for+Heterogeneous+Workloads+in+a+Cloud+Datacenter&author=Garg,+S.K.&author=Toosi,+A.N.&author=Gopalaiyengar,+S.K.&author=Buyya,+R.&publication_year=2014&journal=J.+Netw.+Comput.+Appl.&volume=45&pages=108%E2%80%93120&doi=10.1016/j.jnca.2014.07.030)] [CrossRef (https://doi.org/10.1016/j.jnca.2014.07.030)]
- 29. Song, W.; Xiao, Z.; Chen, Q.; Luo, H. Adaptive Resource Provisioning for the Cloud Using Online Bin Packing. *IEEE Trans. Comput.* **2014**, *63*, 2647–2660. [Google Scholar (https://scholar.google.com/scholar_lookup?title=Adaptive+Resource+Provisioning+for+the+Cloud+Using+Online+Bin+Packing&author=Song,+W.&author=Xiao,+Z.&author=Chen,+Q.&author=Luo,+H.&publication_year=2014&journal=IEEE+Trans.+Comput.&volume=63&pages=2647%E2%80%932660&doi=10.1109/TC.2013.148)] [CrossRef (https://doi.org/10.1109/TC.2013.148)]

-  Quang-Hung, N.; Nien, P.D.; Nam, N.H.; Huynh Tuong, N.; Thoai, N. A Genetic Algorithm for Power-Aware Virtual Machine Allocation in Private Cloud. In Proceedings of the Information and Communication Technology: International Conference, ICT-EurAsia 2013, Yogyakarta, Indonesia, 25–29 March 2013; pp. 183–191. [Google Scholar] (https://scholar.google.com/scholar_lookup?title=A+Genetic+Algorithm+for+Power-Aware+Virtual+Machine+Allocation+in+Private+Cloud&conference=Proceedings+of+the+Information+and+Communication+Technology:+International+Conference+ICT-EurAsia+2013&author=Quang-Hung,+N.+&author=Nien,+P.D.+&author=Nam,+N.H.+&author=Huynh+Tuong,+N.+&author=Thoai,+N.+&publication_year=2013&pages=183%E2%80%93191)]
-  Hung, N. & author=Nien, +P.D.&author=Nam,+N.H.&author=Huynh+Tuong,+N.&author=Thoai,+N.&publication_year=2013&pages=183%E2%80%93191)]
-  Madhusudhan, B.; Sekaran, K.C. A Genetic Algorithm Approach for Virtual Machine Placement in Cloud. In Proceedings of the International Conference on Emerging Computing, Information, Communication and Applications (ERCICA2013), Bangalore, India, 2–3 August 2013. [Google Scholar] ([https://scholar.google.com/scholar_lookup?title=A+Genetic+Algorithm+Approach+for+Virtual+Machine+Placement+in+Cloud&conference=Proceedings+of+the+International+Conference+on+Emerging+Research+in+Computing,+Information,+Communication+and+Applications+\(ERCICA2013\)&author=Madhusudhan,+B.+&author=Sekaran,+K.C.+&publication_year=2013](https://scholar.google.com/scholar_lookup?title=A+Genetic+Algorithm+Approach+for+Virtual+Machine+Placement+in+Cloud&conference=Proceedings+of+the+International+Conference+on+Emerging+Research+in+Computing,+Information,+Communication+and+Applications+(ERCICA2013)&author=Madhusudhan,+B.+&author=Sekaran,+K.C.+&publication_year=2013))]

32. Priya, B.; Pilli, E.S.; Joshi, R.C. A Survey on Energy and Power Consumption Models for Greener Cloud. In Proceedings of the 2013 3rd IEEE International Advance Computing Conference (IACC), Ghaziabad, India, 22–23 February 2013; pp. 76–82. [Google Scholar] ([https://scholar.google.com/scholar_lookup?title=A+Survey+on+Energy+and+Power+Consumption+Models+for+Greener+Cloud&conference=Proceedings+of+the+2013+3rd+IEEE+International+Advance+Computing+Conference+\(IACC\)&author=Priya,+B.+&author=Pilli,+E.S.+&author=Joshi,+R.C.+&publication_year=2013&pages=76%E2%80%9382](https://scholar.google.com/scholar_lookup?title=A+Survey+on+Energy+and+Power+Consumption+Models+for+Greener+Cloud&conference=Proceedings+of+the+2013+3rd+IEEE+International+Advance+Computing+Conference+(IACC)&author=Priya,+B.+&author=Pilli,+E.S.+&author=Joshi,+R.C.+&publication_year=2013&pages=76%E2%80%9382))]

- Syed, Abdul, S.; Malwade, S.; Nursetyo, A.A.; Sood, M.; Bhatia, M.; Barsasella, D.; Liu, M.; Chang, C.-C.; Srinivasan, K.; Li, Y.-C.J.; et al. Virtual Reality among the Elderly: A Usefulness and Acceptance Study from Taiwan. *BMC Geriatr.* **2019**, *19*, 223. [Google Scholar (https://scholar.google.com/scholar_lookup?title=Virtual+Reality+among+the+Elderly:+A+Usefulness+and+Acceptance+Study+from+Taiwan&author=Syed-Abdul+S.&author=Malwade,+S.&author=Nursetyo,+A.A.&author=Sood,+M.&author=Bhatia,+M.&author=Barsasella,+D.&author=Liu,+M.F.&author=Chang,+C.-C.&author=Srinivasan,+K.&author=Li,+Y.-C.J.&publication_year=2019&journal=BMC+Geriatr.&volume=19&pages=223&doi=10.1186/s12877-019-1218-8)] [CrossRef (https://doi.org/10.1186/s12877-019-1218-8)] [Green Version (https://bmcgeriatr.biomedcentral.com/track/pdf/10.1186/s12877-019-1218-8)]
- Talwani, S.; Singla, J. Enhanced Bee Colony Approach for reducing the energy consumption during VM migration in cloud computing environment. *IOP Conf. Ser. Mater. Sci. Eng.* **2021**, *1022*, 012069. [Google Scholar (https://scholar.google.com/scholar_lookup?title=Enhanced+Bee+Colony+Approach+for+reducing+the+energy+consumption+during+VM+migration+in+cloud+computing+environment&author=Talwani,+S.&author=Singla,+J.&publication_year=2021&journal=IOP+Conf.+Ser.+Mater.+Sci.+Eng.&volume=1022&pages=012069&doi=10.1088/1757-899X/1022/1/012069)] [CrossRef (https://doi.org/10.1088/1757-899X/1022/1/012069)]

35. Talwani, S.; Singla, J. Enhanced Bee Colony Approach for reducing the energy consumption during VM migration in cloud computing environment. *IOP Conf. Ser. Mater. Sci. Eng.* **2021**, *1022*, 012069. [Google Scholar (https://scholar.google.com/scholar_lookup?title=Enhanced+Bee+Colony+Approach+for+reducing+the+energy+consumption+during+VM+migration+in+cloud+computing+environment&author=Talwani,+S.&author=Singla,+J.&publication_year=2021&journal=IOP+Conf.+Ser.+Mater.+Sci.+Eng.&volume=1022&pages=012069&doi=10.1088/1757-899X/1022/1/012069)] [CrossRef (https://doi.org/10.1088/1757-899X/1022/1/012069)]

- [Download PDF \(1424-8220/23/13/6117/pdf?version=1688376187\)](#)
 - [Introduction](#)
 - [Share](#)
 - [Literature Survey](#)
 - [Motivation of the Research](#)
 - [Help](#)
 - [Methodology](#)
 - [Results and Discussion](#)
 - [Cite](#)
 - [Conclusions](#)
 - [Author Contributions](#)
 - [Funding](#)
 - [Supervision](#)
 - [Institutional Review Board](#)
 - [Statement of Ethics](#)
 - [Data Availability Statement](#)
 - [Acknowledgments](#)
 - [Conflicts of Interest](#)
 - [References](#)
- Dai, X.; Xiao, Z.; Jiang, H.; Alazab, M.; Lui, J.C.; Min, G.; Liu, J. Task offloading for cloud assisted fog computing with dynamic service caching in enterprise management systems. *IEEE Trans. Ind. Inform.* **2022**, *19*, 662–672. [Google Scholar (https://scholar.google.com/scholar_lookup?title=Task+offloading+for+cloud-assisted+fog+computing+with+dynamic+service+caching+in+enterprise+manag+ement+systems&author=Dai,+X.&author=Xiao,+Z.&author=Jiang,+H.&author=Alazab,+M.&author=Lui,+J.C.&author=Min,+G.&author=Liu,+J.&publication_year=2022&journal=IEEE+Trans.+Ind.+Inform.&volume=19&pages=662%E2%80%93672&doi=10.1109/TII.2022.3186641)] [CrossRef (<https://doi.org/10.1109/TII.2022.3186641>)]
- Tran, C.H.; Bui, T.K.; Pham, T.V. Virtual machine migration policy for multi-tier groups public. *Comput. Intell.* **2022**, *36*, 1285–1306. [Google Scholar (https://scholar.google.com/scholar_lookup?title=Virtual+machine+migration+policy+for+multi-tier+groups+public&author=Tran,+C.H.&author=Bui,+T.K.&author=Pham,+T.V.&publication_year=2022&journal=Comput.+Intell.&volume=36&pages=1285%E2%80%931306&doi=10.1007/s00607-021-01047-0)] [CrossRef (<https://doi.org/10.1007/s00607-021-01047-0>)]

38. Abedi, S.; Ghobaei-Arani, M.; Khorami, E.; Mojarad, M. Dynamic Resource Allocation Using Improved Firefly Optimization Algorithm in Cloud Environment. *Appl. Artif. Intell.* **2022**, *36*, 2055394. [Google Scholar (https://scholar.google.com/scholar_lookup?title=Dynamic+Resource+Allocation+Using+Improved+Firefly+Optimization+Algorithm+in+Cloud+Environment&author=Abedi,+S.&author=Ghobaei-Arani,+M.&author=Khorami,+E.&author=Mojarad,+M.&publication_year=2022&journal=Appl.+Artif.+Intell.&volume=36&pages=2055394&doi=10.1080/08839514.2022.2055394)] [CrossRef (<https://doi.org/10.1080/08839514.2022.2055394>)]
39. Khan, M.S.A.; Santhosh, R. Hybrid Optimization Algorithm for VM Migration in Cloud Computing. *Comput. Electr. Eng.* **2022**, *102*, 108152. [Google Scholar (https://scholar.google.com/scholar_lookup?title=Hybrid+Optimization+Algorithm+for+VM+Migration+in+Cloud+Computing&author=Khan,+M.S.A.&author=Santhosh,+R.&publication_year=2022&journal=Comput.+Electr.+Eng.&volume=102&pages=108152&doi=10.1016/j.compeleceng.2022.108152)] [CrossRef (<https://doi.org/10.1016/j.compeleceng.2022.108152>)]

- Zhao, H.; Wang, Y.; Han, X.; Jin, S. MAP based modeling method and performance study of a task offloading scheme with time-correlated traffic and VM repair in MEC systems. *Wirel. Netw.* **2023**, *29*, 47–68. [Google Scholar] (https://scholar.google.com/scholar_lookup?title=MAP+based+modeling+method+and+performance+study+of+a+task+offloading+scheme+with+time-correlated+traffic+and+VM+repair+in+MEC+systems&author=Zhao+H+Wang,+Y.&author=Han,+X.&author=Jin,+S.&publication_year=2023&journal=Wirel.+Netw.&volume=29&pages=47%E2%80%9368&doi=10.1007/s11276-022-03099-2)
 - Singh, N.; Hamid, Y.; Juneja, S.; Srivastava, G.; Dhiman, G.; Gadekallu, T.R.; Shah, M.A. Load balancing and service discovery using Docker Swarm for microservice based big data applications. *J. Cloud Comput.* **2023**, *12*, 4. [Google Scholar] (https://scholar.google.com/scholar_lookup?title=Load+balancing+and+service+discovery+using+Docker+Swarm+for+microservice+based+big+data+applications&author=Singh,+N.&author=Hamid,+Y.&author=Juneja,+S.&author=Srivastava,+G.&author=Dhiman,+G.&author=Gadekallu,+T.R.&author=Shah,+M.A.&publication_year=2023&journal=J.+Cloud+Comput.&volume=12&pages=4&doi=10.1186/s13677-022-00358-7)
42. Singh, N.; Hamid, Y.; Juneja, S.; Srivastava, G.; Dhiman, G.; Gadekallu, T.R.; Shah, M.A. Load balancing and service discovery using Docker Swarm for microservice based big data applications. *J. Cloud Comput.* **2023**, *12*, 4. [Google Scholar] (https://scholar.google.com/scholar_lookup?title=Load+balancing+and+service+discovery+using+Docker+Swarm+for+microservice+based+big+data+applications&author=Singh,+N.&author=Hamid,+Y.&author=Juneja,+S.&author=Srivastava,+G.&author=Dhiman,+G.&author=Gadekallu,+T.R.&author=Shah,+M.A.&publication_year=2023&journal=J.+Cloud+Comput.&volume=12&pages=4&doi=10.1186/s13677-022-00358-7)

- Kavitha, C.; Gadekallu, T.R.; Kavin, B.P.; Lai, W.C. Filter-Based Ensemble Feature Selection and Deep Learning Model for Intrusion Detection in Cloud Computing. *Electronics* **2023**, *12*, 556. [Google Scholar] (https://scholar.google.com/scholar_lookup?title=Filter-Based+Ensemble+Feature+Selection+and+Deep+Learning+Model+for+Intrusion+Detection+in+Cloud+Computing&author=Kavitha,+C.&author=Gadekallu,+T.R.&author=Kavin,+B.P.&author=Lai,+W.C.&publication_year=2023&journal=Electronics&volume=12&pages=556&doi=10.3390/electronics12030556) [CrossRef] (<https://doi.org/10.3390/electronics12030556>)
- Zhang, G.; Wang, J.; Wang, Q.; Wan, B. VM performance-aware virtual machine migration method based on ant colony optimization in cloud environment. *IEEE J. Sel. Top. Signal Process.* **2023**, *17*, 27–37. [Google Scholar] (https://scholar.google.com/scholar_lookup?title=VM+performance-aware+virtual+machine+migration+method+based+on+ant+colony+optimization+in+cloud+environment&author=Zhang,+G.&author=Wang,+J.&author=Wang,+Q.&author=Wan,+B.&publication_year=2023&journal=IEEE+J.+Sel.+Top.+Signal+Process.&volume=17&pages=27%2F37&doi=10.1109/JSTSP.2023.3221271) [CrossRef] (<https://doi.org/10.1109/JSTSP.2023.3221271>)
- 45. Wu, Q.; Zhao, Y.; Fan, Q.; Fan, P.; Wang, J.; Zhang, C. Mobility-Aware Cooperative Caching in Vehicular Edge Computing Based on Asynchronous Federated and Deep Reinforcement Learning. *IEEE J. Sel. Top. Signal Process.* **2023**, *17*, 66–81. [Google Scholar] (https://scholar.google.com/scholar_lookup?title=Mobility-Aware+Cooperative+Caching+in+Vehicular+Edge+Computing+Based+on+Asynchronous+Federated+and+Deep+Reinforcement+Learning&author=Wu,+Q.&author=Zhao,+Y.&author=Fan,+Q.&author=Fan,+P.&author=Wang,+J.&author=Zhang,+C.&publication_year=2023&journal=IEEE+J.+Sel.+Top.+Signal+Process.&volume=17&pages=66%2F81&doi=10.1109/JSTSP.2022.3221271) [CrossRef] (<https://doi.org/10.1109/JSTSP.2022.3221271>)

-  Song, Y.; Xin, R.; Chen, P.; Zhang, R.; Chen, J.; Zhao, Z. Identifying performance anomalies in fluctuating cloud environments. *Future Gener. Comput. Syst.* **2023**, *145*, 77–86. [Google Scholar] (https://scholar.google.com/scholar_lookup?title=Identifying+performance+anomalies+in+fluctuating+cloud+environments:+A+robust+correlative-GNN-based+explainable+approach&author=Song,+Y.&author=Xin,+R.&author=Chen,+P.&author=Zhang,+R.&author=Chen,+J.&author=Zhao,+Z.&publication_year=2023&journal=Future+Gener.+Comput.+Syst.&volume=145&pages=77%E2%80%9386&doi=10.1016/j.future.2023.03.020) [CrossRef] (<https://doi.org/10.1016/j.future.2023.03.020>)
-  Jiang, H.; Dai, X.; Xiao, Z.; Iyengar, A. K. Joint Task Offloading and Resource Allocation for Energy-Constrained Mobile Edge Computing. *IEEE Trans. Mob. Comput.* **2022**, *22*, 4000–4015. [Google Scholar] (https://scholar.google.com/scholar_lookup?title=Joint+Task+Offloading+and+Resource+Allocation+for+Energy-Constrained+Mobile+Edge+Computing&author=Jiang,+H.&author=Dai,+X.&author=Xiao,+Z.&author=Iyengar,+A.K.&publication_year=2022&journal=IEEE+Trans.+Mob.+Comput.&volume=22&pages=4000%E2%80%934015&doi=10.1109/TMC.2022.3150432) [CrossRef] (<https://doi.org/10.1109/TMC.2022.3150432>)

48. Uppal, M.; Gupta, D.; Juneja, S.; Dhiman, G.; Kautish, S. Cloud-based fault prediction using IoT in office automation for improvisation of health of employees. *J. Healthc. Eng.* **2021**, *2021*, 8106467. [Google Scholar] (https://scholar.google.com/scholar_lookup?title=Cloud-based+fault+prediction+using+IoT+in+office+automation+for+improvisation+of+health+of+employees&author=Uppal,+M.&author=Gupta,+D.&author=Juneja,+S.&author=Dhiman,+G.&author=Kautish,+S.&publication_year=2021&journal=J.+Healthc.+Eng.&volume=2021&pages=8106467&doi=10.1155/2021/8106467) [CrossRef] (<https://doi.org/10.1155/2021/8106467>)

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

© 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>) (<https://creativecommons.org/licenses/by/4.0/>).

ADRI (A)
Down


<mailto:?>

Literature Survey

& Subject From MDPI Algorithmic Approaches to Virtual Machine Migration of the Research Cloud Computing with Updated SESA Algorithm Body=<https://www.mdpi.com/2371550>**%3A%0A%0AAlgorithmic Approaches to Virtual Machine Migration in Cloud Computing with Updated SESA Algorithm Abstract Cloud computing Important role in every IT sector. Many tech giants such as Google C Microsoft C and Facebook C as Institutions Review Board Data centres around the world to provide computational and storage services. The customers either directly or indirectly use the service. The submission of the jobs to the cloud centres. The preliminary aim is to reduce the overall consumption which was ignored in the early days of cloud development. This was due to the performance expectations from cloud servers as they were supposed to provide all the services through their services layers IaaS PaaS and SaaS. As time passed and researchers came up with new terminologies and algorithmic architecture for the reduction of power consumption and sustainability other algorithmic anarchies were also introduced such as statistical oriented learning and bioinspired algorithms. In this paper an indepth focus has been done on multiple approaches for migration among virtual machines and find out various issues among existing approaches. The proposed work utilizes elastic scheduling inspired by the smart elastic scheduling algorithm SE-SA to develop a more energy-efficient VM[...]**

text=Algorithmic+Approach+to+Virtual+Machine+Migration+in+Cloud+Computing+with
+Updated+SESA+Algorithm&hashtags=mdpisensors&url=https%3A%2F%2Fwww.mdpi
.com%2F2371550&via=Sensors_MDPI)  ([http://www.linkedin.com/shareArticle?](http://www.linkedin.com/shareArticle?mini=true&url=https%3A%2F%2Fwww.mdpi.com%2F2371550&title=Algorithmic%20Approach%20to%20Virtual%20Machine%20Migration%20in%20Cloud%20Computing%20with%20Updated%20SESA%20Algorithm%26source%3Dhttps%3A%2F%2Fwww.mdpi.com%26summary%3DCloud%20computing%20plays%20an%20important%20role%20in%20every%20IT%20sector.%20Many%20tech%20giants%20such%20as%20Google%2C)
mini=true&url=https%3A%2F%2Fwww.mdpi.com%2F2371550&title=Algorithmic%20Ap
proach%20to%20Virtual%20Machine%20Migration%20in%20Cloud%20Computing%20
with%20Updated%20SESA%20Algorithm%26source%3Dhttps%3A%2F%2Fwww.mdpi.c
om%26summary%3DCloud%20computing%20plays%20an%20important%20role%20in
%20every%20IT%20sector.%20Many%20tech%20giants%20such%20as%20Google%2C

36/39

%20Microsoft%2C%20and%20Facebook%20as%20deploying%20their%20data%20cent

res%20around%20the%20world%20to%20provide%20computation%20and%20storage

%20services.%20The%20customers%20either%20submit%20their%20job%20%5B...%5

D)Motivation of the Research

(https://www.facebook.com/sharer.php?u=https://www.mdpi.com/2371550)

(http://www.reddit.com/submit?url=https://www.mdpi.com/2371550)

(https://www.mendeley.com/import/?url=https://www.mdpi.com/2371550)

Conclusions

MDPI and ACS Style

Author Contributions

Kaur, A.; Kumar, S.; Gupta, D.; Hamid, Y.; Hamdi, M.; Ksibi, A.; Elmannai, H.; Saini, S.

Algorithmic Approach to Virtual Machine Migration in Cloud Computing with Updated SESA

Algorithm. *Sensors* **2023**, *23*, 6117. <https://doi.org/10.3390/s23136117>

groups/public/10.3390/s23136117?

AMM Style Consent Statement

Data Availability Statement

Kaur, A.; Kumar, S.; Gupta, D.; Hamid, Y.; Hamdi, M.; Ksibi, A.; Elmannai, H.; Saini, S. Algorithmic

Approach to Virtual Machine Migration in Cloud Computing with Updated SESA Algorithm.

Sensors **2023**, *23*, 6117. <https://doi.org/10.3390/s23136117>

References

Chicago/Turabian Style

Kaur, Amandeep, Saurabh Kumar, Deepali Gupta, Yasir Hamid, Monia Hamdi, Amel Ksibi,

Hela Elmannai, and Shilpa Saini. 2023. "Algorithmic Approach to Virtual Machine Migration in

Cloud Computing with Updated SESA Algorithm" *Sensors* 23, no. 13: 6117.

<https://doi.org/10.3390/s23136117>

ⓘ Note that from the first issue of 2016, this journal uses article numbers instead of page

numbers. See further details [here](#)

(<https://www.mdpi.com/about/announcements/784>).

Article Metrics

Citations

Crossref

ads

Web of
Science

Scopus

Google
Scholar

10

(<https://ui.adsa>

GWVersion=28

11 (<https://>

partnerID=HzO

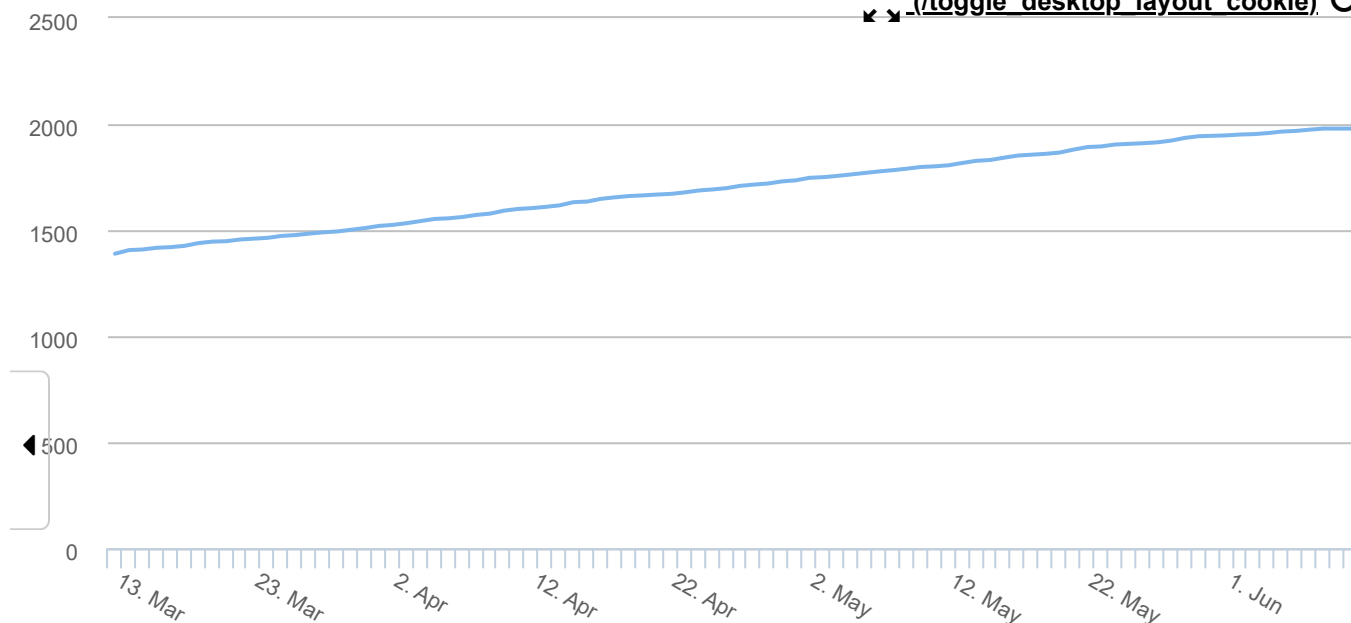
title=Algorithmi

Back to TopTop

Article Access Statistics

[Download PDF \(/1424-8220/23/13/6117/pdf?version=1688376187\)](#)

Article access statistics

[Order Article Reprints \(/1424-8220/23/13/6117/reprints\)](#)
[toggle desktop layout cookie](#)


Article Views

For more information on the journal statistics, click [here \(/journal/sensors/stats\)](/journal/sensors/stats).

ⓘ Multiple requests from the same IP address are counted as one view.

[Sensors \(/journal/sensors\)](/journal/sensors), EISSN 1424-8220, Published by MDPI

[RSS \(/rss/journal/sensors\)](/rss/journal/sensors) **[Content Alert \(/journal/sensors/toc-alert\)](/journal/sensors/toc-alert)**

Further Information

[Article Processing Charges \(/apc\)](/apc)

[Pay an Invoice \(/about/payment\)](/about/payment)

[Open Access Policy \(/openaccess\)](/openaccess)

[Contact MDPI \(/about/contact\)](/about/contact)

[Jobs at MDPI \(https://careers.mdpi.com\)](https://careers.mdpi.com)

Guidelines

[For Authors \(/authors\)](/authors)

[For Reviewers \(/reviewers\)](/reviewers)

[For Editors \(/editors\)](/editors)

[For Librarians \(/librarians\)](/librarians)

[For Publishers \(/publishing_services\)](/publishing_services)

[For Societies \(/societies\)](/societies)

[Back to Top](#)

For Conference Organizers ([/conference_organizers](#))

MDPI Open Access PDF ([/1424-8220/23/13/6117/pdf?version=1688376187](#))

Introduction

Sciforum (<https://sciforum.net>)

Literature Survey

MDPI Books (<https://www.mdpi.com/books>)

Motivation of the Research

Preprints.org (<https://www.preprints.org>)

Methodology

Scilit (<https://www.scilit.net>)

Results and Discussion

SciProfiles (<https://sciprofiles.com>)

Conclusions

[utm_source=mdpi.com&utm_medium=bottom_menu&utm_campaign=initiative](#)

Author Contributions

Encyclopedia (<https://encyclopedia.pub>)

Discussion

JAMIS (<https://jams.pub>)

<https://sciprofiles.com/discussion->

[Statement Series \(/about/proceedings\)](#)

[groups/public/10.3390/S23136117](#)

[utm_source=mdpi.com&utm_medium=publication&utm_campaign=discuss_in_sciprofiles](#)

Informed Consent Statement

Data Availability Statement

Follow MDPI

Acknowledgments

LinkedIn (<https://www.linkedin.com/company/mdpi>)

Conflicts of Interest

Facebook (<https://www.facebook.com/MDPIOpenAccessPublishing>)

References

Twitter (<https://twitter.com/MDPIOpenAccess>)

Comment

Order Article Reprints ([/1424-8220/23/13/6117/rex](#))

[\(/toggle desktop layout cookie\)](#) 🔍 ☰

Subscribe to receive issue release
notifications and newsletters from
MDPI journals

Select options ▼

Enter your email address...

Subscribe

© 1996-2024 MDPI (Basel, Switzerland) unless otherwise stated

[Disclaimer](#) [Terms and Conditions \(/about/terms-and-conditions\)](#)

[Privacy Policy \(/about/privacy\)](#)

Back to TopTop